

## KARTA PRZEDMIOTU PSW

Nazwa przedmiotu	<b>Programowanie systemów wbudowanych</b>
Prowadzący wykład	mgr inż. Mariusz Rudnicki, dr inż. Krzysztof Czarnecki
Prowadzący projekt	mgr inż. Mariusz Rudnicki, dr inż. Krzysztof Czarnecki, dr inż. Iwona Kochańska
Kierunek	AiR, EiT, IBM, Inf.
Specjalność	Systemy wbudowane

## Wymiar tygodniowy

Stopień	Semestr	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium
2	1	20/15	-	-	1	-

## Cele przedmiotu

**Wykład:** Zapoznanie studentów z budową i działaniem systemów operacyjnych oraz programów układowych i rozruchowych dla systemów wbudowanych, możliwościami ich modyfikacji dla poprawy wydajności oraz rozszerzenia funkcjonalności o nowe usługi i sterowniki.

**Projekt:** Zapoznanie z metodyką skutecznego projektowania oprogramowania systemów wbudowanych (firmware).

Wykład

Zagadnienia	poziom					liczba godzin
	wiedzy			umiej		
	A	B	C	D	E	
<b>1. Systemy operacyjne dla systemów wbudowanych</b>						<b>5</b>
1.1. Standard POSIX	A					0,33
1.2. Systemy operacyjne w systemach wbudowanych (Yocto, Debian)		B				0,67
1.3. Systemy operacyjne czasu rzeczywistego (QNX Neutrino, RT Linux, Windows Embedded, ReMix)	A					0,67
1.4. Jądro i jego otoczenie w systemach operacyjnych RT/systemach wbudowanych		B				0,67
1.5. Manager procesów. Zarządzanie przestrzenią nazw. Zarządzanie pamięcią		B				0,67
1.6. Wątki i procesy. Algorytmy szeregowania wątków. Metody synchronizacji wątków. Komunikacja międzyprocesowa.			C			0,67
1.7. Koncepcje obsługi przerwań sprzętowych			C			0,66
1.8. System plików		B				0,66
<b>2. System operacyjny "uszyty na miarę" platformy sprzętowej</b>						<b>5</b>
2.1. Instalacja OS	A					0,33
2.2. Obraz systemu operacyjnego		B				0,33
2.3. Sposoby bootowania OS		B				0,67
2.4. Kolejność uruchamiania sterowników i usług		B				0,67
2.5. Konfiguracja OS - ograniczenie funkcjonalności		B				1
2.6. Rozszerzenie funkcjonalności OS o nowe usługi		B				1
2.7. Sterowniki			C			1
<b>3. Oprogramowanie układowe (firmware) – jeszcze nie OS</b>						<b>3</b>
3.1. Boot Loader	A					0,33
3.2. Bloki funkcjonalne boot loader'a (moduł głównych funkcji, moduł sterownika I/O, moduł sterownika pamięci, główne funkcje, sterownik I/O,		B				0,34
3.3. BSPs - Board Support Packages	A					0,34

3.4. Kolejność uruchamiania sterowników			C			0,33
3.5. Projektowanie oprogramowania układowego		B				0,33
3.6. Modyfikacja i testowanie oprogramowania układowego		B				0,67
3.7. BIOS		B				0,33
3.8. UEFI		B				0,33
<b>4. Tworzenie oprogramowania dla systemów wbudowanych</b>						<b>7</b>
4.1. Tworzenie oprogramowania systemów wbudowanych (języki programowania - C, Python)			C			2
4.2. Problemy tworzenia oprogramowania wielowątkowego, współbieżność procesów, reguły dostępu do zasobów wspólnych, systemy blokad i zarządzanie nimi.			C			2
4.3. Problem poprawności wykonania współbieżnego zadań, kryteria szeregowności zadań, przykładowe algorytmy sprawdzania szeregowności zadań.			C			1,67
4.4. Wskaźniki wydajności: throughput, execution time, wait time		B				0,33
4.5. Techniki szybkiego programowania (Simulink)		B				1
<i>Razem:</i>						<b>20</b>